Analytical model of two level scheduling algorithm for WiMAX networks

Zeeshan Ahmed, Salima Hamma

Analytical Model of Two Level Scheduling Algorithm for WiMAX Networks

Zeeshan Ahmed
Department of Computing and Technology,
Indus University, Karachi, Pakistan
zeeshan.ahmed@indus.edu.pk

Salima Hamma
LUNAM Université IRCCyN,
CNRS UMR 6597, Polytech′ Nantes,
rue Christian Pauc - BP 50609 -
44306 Nantes Cedex 3 France
salima.hamma@univ-nantes.fr

***Abstract*—Two Level Scheduling Algorithm (TLSA) is a QoS-enabled fair and efficient connection admission control and packet scheduling algorithm for WiMAX networks. At the first level, an inter-class scheduling algorithm distributes bandwidth among various WiMAX service classes. Then at the second level, class specific algorithms distribute bandwidth among connections of the associated class. The present paper focuses on a Markov chain based analytical model of TLSA that is comprehensive enough to depict the behavior of inter-class and intra-class scheduling algorithms. Extensive simulations were performed and several criteria were considered to assess the accuracy of the proposed model. We considered bandwidth allocation in both inter-class and intra-class scheduling algorithms, percentage of lost packets and the service ratio. The experiments indicate that the analytical model faithfully captures the behavior of TLSA.**

I. Introduction

Two-level Scheduling Algorithm (TLSA) [3] is a fair and efficient connection admission control (CAC) and packet scheduling scheme for IEEE 802.16e [7] networks. Packet scheduling and connection admission control (CAC) are two of the most important functions of a QoS framework in communication networks. In 802.16 point-to-multipoint network, a base station (BS) provides services to multiple subscriber stations (SS).

In the 802.16 standard, the complex task of packet scheduling is distributed among three schedulers, i.e. BS uplink scheduler, BS downlink scheduler and SS scheduler. In the uplink direction, packet scheduling is done by a cooperation of the BS uplink scheduler and the SS scheduler. In 802.16, a CAC module at the BS facilitates the working of packet schedulers by selectively admitting new connections to the network. The 802.16 standard does not specify algorithms for CAC module and packet schedulers.

TLSA was proposed as a simple, efficient and practical CAC and packet scheduling scheme for 802.16 networks. The main objective is to fairly distribute bandwidth among various connections while guaranteeing QoS. The algorithm is fast enough to support very high data rates, and therefore it is suitable for 802.16 networks.

TLSA consists of two levels to efficiently furnish QoS to various service classes. The details of WiMAX service classes can be found in [7]. At the first level an inter-class algorithm distributes bandwidth among various classes of traffic according to their bandwidth demands and QoS specifications. Then at the second level, each class is treated by an intra-class scheduling algorithm. An intra-class scheduling algorithm takes the bandwidth allocated by the inter-class scheduling algorithm and distributes it among connections of the associated class. The complete working of TLSA has been presented in [2] and [3].

In this paper we provide an analytical model of TLSA. Due to hierarchical structure of TLSA, the development of an accurate model is not trivial. The model provided in the paper is based on in-depth queuing analysis of TLSA and it encompasses all classes of services supported by WiMAX.

The rest of the paper is organized as follows. Section II provides a detailed structure and working of TLSA. The analytical model is then presented in Section III. Section IV discusses the simulation results that were performed to assess the validity of the model. Finally, Section V concludes the paper.

II. Two-Level Scheduling Algorithm

This section presents the working of TLSA. The section begins by providing the working of CAC module. Then Section II-C1 gives the details of inter-class scheduling algorithm. Finally, the details of each intra-class scheduling algorithm are presented in Section II-C2.

A. Packet Size and Deadline Estimation

For correct operation, TLSA must know the size of traffic of each connection arrived during the previous MAC frame. An SS sends aggregate bandwidth request and the BS uplink scheduler does not know the size and deadlines of individual packets. Therefore, the BS has to estimate these parameters. Arrival-service curve [8] provides a convenient way to determine the size of data traffic arrived during previous MAC frame.

Let $\varrho_i[f]$ be the backlog of connection i at the start of frame f and $\Upsilon_i[f-1]$ is the service received by i in frame $f-1$. Then the traffic ($\zeta_i[f-1]$) arrived during frame $f-1$ can be determined by Equation 1.

$$\zeta_i[f-1] = (\varrho_i[f] - \varrho_i[f-1]) + \Upsilon_i[f-1] \qquad (1)$$

The realtime packets that misses their deadline are dropped from scheduling queues. Therefore, Equation 1 must be modified to take into account the dropped packets. If $d_i[f-1]$ is the traffic dropped during $f-1$, then the size of traffic arrived during $f-1$ can be determined by Equation 2.

$$\zeta_i[f-1] = (\varrho_i[f] - \varrho_i[f-1]) + \Upsilon_i[f-1] + d_i[f-1] \quad (2)$$

The algorithm estimates the deadlines of packets by taking into account the maximum tolerable latency of associated variable bit-rate realtime connections. If δ_i is the maximum latency of connection i and γ is the duration of MAC frame, then the packets arrived during frame $f-1$ must be scheduled before frame $f-1+\frac{\delta_i}{\gamma}$ to avoid expiry of deadline.

B. Connection Admission Control (CAC)

The decision of CAC is based on the QoS specifications of both existing and new connections, and the available uplink bandwidth. The proposed algorithm accepts a new connection if the following conditions are satisfied: (i) The requested QoS can be provided to the new connection (ii) QoS guarantees of existing connections are not breached.

The upper bound on latency of an rtPS connection i can be guaranteed if condition 3 holds true [3].

$$\alpha''_i \le \left(\frac{\delta_i}{\gamma} - 1\right)\left((\beta - \beta') - \Psi' \sum_{j\epsilon\Delta_{rtPS}-\{i\}} \alpha'_j\right) \quad (3)$$

where α''_i is the maximum sustained traffic rate of connection i, β is the total uplink capacity, Ψ' is mean service ratio as determined by equation 10, Δ_{rtPS} is the set of all admitted connections of rtPS class, α'_j is the average traffic rate of connection j and

$$\beta' = \sum_{k\epsilon\Delta_{UGS}} \alpha_k + \sum_{m\epsilon\Delta_{ertPS}} \alpha_m + \sum_{n\epsilon\Delta_{nrtPS}} \alpha_n + \beta_{BE}$$

where α_v is the minimum reserved traffic rate (MRTR) of connection v and β_{BE} is the bandwidth reserved for BE class. The class-wise operation of the proposed CAC algorithm is given below.

1) Best-Effort (BE) Class: A BE connection does not have QoS requirements. Therefore, the proposed algorithm always admit a new BE connection.

2) Non-Realtime Polling Service (nrtPS): An nrtPS connection requires guaranteed minimum traffic rate. CAC admits an nrtPS connection, if the minimum traffic rate requested by the connection is less than or equal to the available uplink bandwidth and the condition specified by Equation 3 is satisfied for each established rtPS connection.

3) Realtime Polling Service (rtPS): An rtPS connection demands guarantees on both minimum traffic rate and maximum delay. A new rtPS connection is admitted if the minimum traffic rate specified by the connection is less than or equal to the available bandwidth and the condition specified by Equation 3 is satisfied for both new and existing rtPS connections.

C. Uplink Packet Scheduling

TLSA is a hierarchical scheduling scheme for BS uplink scheduler. At the first level, an inter-class scheduling algorithm distributes bandwidth among various service classes according to their bandwidth demands, QoS specifications and available network resources. Then at the second level, each class has an associated scheduling algorithm that distributes bandwidth among connections of that class.

1) Inter-Class Scheduling: The BS uplink scheduler manages unique queues for holding bandwidth requests for each class of traffic. The algorithm enforces service priority by the order of processing of scheduling queues. The queues are processed in the order rtPS, nrtPS, and BE. Thus, rtPS class has the highest priority while BE class has the lowest priority.

The algorithm makes fixed bandwidth allocation on periodic basis to Unsolicited Grant Service (UGS) and Enhanced-rtPS (ertPS) classes, as specified by the 802.16 standard. Therefore, subsequently we would discuss scheduling of rtPS, nrtPS and BE classes only.

To prevent starvation of BE class, the algorithm reserves a fixed part of uplink bandwidth, which is denoted by β_{BE}. The value of β_{BE} is not fixed and could be adjusted on per frame basis. However, the algorithm makes sure that the condition given by inequality 4 is always satisfied.

$$\beta_{BE} \le \sum_{j\epsilon\Delta_{BE}} \varrho_j[f] \; and \; \beta_{BE} \le \iota \quad (4)$$

where $\varrho_i[f]$ is the backlog of connection i at the start of frame f. The value of ι is specified by service providers to suit their business models.

To nrtPS class, the algorithm first allocates enough bandwidth so that the minimum traffic rate of each nrtPS connection could be ensured. The minimum bandwidth allocated to nrtPS class (β_{nrtPS}) is calculated by Equation 5.

$$\beta_{nrtPS} = \sum_{i\epsilon\Delta_{nrtPS}} min\left(\alpha_i, \varrho_i[f]\right) \quad (5)$$

Since β_{nrtPS} and β_{BE} are reserved bandwidths, therefore the bandwidth available for rtPS class is equal to $\beta - \beta_{nrtPS} - \beta_{BE}$. However, actual bandwidth allocation (Θ_{rtPS}) depends upon the bandwidth requests stored in rtPS queues, and it is determined by Equation 6.

$$\Theta_{rtPS}[f] = \sum_{k\epsilon\Delta_{rtPS}} min\left(\varrho_k[f], \beta - (\alpha'_j - \beta_{nrtPS} - \beta_{BE})\right) \quad (6)$$

The unused bandwidth after these allocations is distributed first to nrtPS and then to BE classes. Thus the total bandwidth allocated to nrtPS class is given by Equation 7 and the maximum bandwidth available for BE class is given by Equation 8.

$$\begin{aligned}\Theta_{nrtPS}[f] = \beta_{nrtPS} + min(&\sum_{i\epsilon\Delta_{nrtPS}} \varrho_i[f] - \beta_{nrtPS}, \\ &\beta - \Theta_{rtPS}[f] - \sum_{j\epsilon\Delta_{UGS}\cup\Delta_{ertPS}} \alpha'_j)\end{aligned} \quad (7)$$

$$\Theta_{BE}[f] = \left(\beta - \Theta_{nrtPS} - \Theta_{rtPS} - \sum_{j \epsilon \Delta_{UGS} \cup \Delta_{ertPS}} \alpha'_j\right) \tag{8}$$

2) Intra-Class Scheduling:

a) rtPS Scheduling: The rtPS intra-class scheduling algorithm ensures QoS for all rtPS connections as well as fairness of bandwidth distribution. To ensure fair bandwidth distribution, the algorithm calculates two parameters, i.e. *Service Ratio*(Ψ_i) and *Mean Service Ratio* (Ψ'), as represented by Equations 9 and 10 respectively.

$$\Psi_i[f] = \frac{\sum_{t=1}^{f-1} \Upsilon_i[t]}{\sum_{t=1}^{f-1} \Gamma_i[t]} \tag{9}$$

where, $\Gamma_i[t]$ is the bandwidth requested by connection i at the start of frame t and $\Upsilon_i[t]$ is the service received by i during t.

$$\Psi'[f] = \frac{\sum_{t=1}^{f-1} \sum_{i \epsilon \Delta_{rtPS}} \Upsilon_i[t]}{\sum_{t=1}^{f-1} \sum_{i \epsilon \Delta_{rtPS}} \Gamma_i[t]} \tag{10}$$

In frame f, an rtPS connection i is eligible to get bandwidth if $\Psi_i[f] \leq \Psi'[f]$. Thus the algorithm takes into account leading and lagging flows [5] by taking bandwidth from leading flows and distributing it among lagging flows. A flow is considered leading, if $\Psi_i[f] > \Psi'$. While for a lagging flow, $\Psi' > \Psi_i[f]$.

When a new bandwidth request $\Gamma_i[f]$ arrives, the algorithm determines whether the connection is eligible to receive uplink bandwidth. If $\Psi_i[f] \leq \Psi'[f]$, then the algorithm tries to fulfill the request in f. However, if the bandwidth available in f is less than $\Gamma_i[f]$, then the algorithm uses the available bandwidth in f to schedule a part of $\Gamma_i[f]$. The remaining part of $\Gamma_i[f]$ is then scheduled in frames $f+1$ to $f + \frac{\delta_i}{\gamma}$. For a detailed explanation of scheduling process, please see [3].

The packets that the algorithm is unable to schedule before $f + \frac{\delta_i}{\gamma}$ are dropped from scheduling queues. Under full bandwidth utilization, the ratio of dropped packets (χ_i) to the total packets generated by an rtPS connection i is independent of data generation rate and it is given by Equation 11 [1].

$$\chi_i = 1 - \left(\frac{\beta - \beta_i}{\sum_{k \epsilon \Delta_{rtPS}} \alpha'_k}\right) \tag{11}$$

b) nrtPS Scheduling: The nrtPS intra-class scheduling algorithm first ensures that the minimum traffic rate requirement is satisfied for each connection. Then any available bandwidth is distributed among needy nrtPS connections in proportion to their backlog. That is an nrtPS connection u first gets $min(\varrho_u[f], \alpha_u)$ units of bandwidth. After this allocation, let the bandwidth requirements of connection u is equal to $\eta_u = \varrho_u[f] - min(\varrho_u[f], \alpha_u)$, and r_n be the bandwidth available for nrtPS class, after satisfying MRTR of each nrtPS connection. Then the total bandwidth allocated to u is given by Equation 12.

$$\Theta_u = min\left(\varrho_u[f], \alpha_u\right) + min\left(r_n, \sum_{v \epsilon \Delta_{nrtPS}} \eta_v\right)\left(\frac{\eta_u}{\sum_{v \epsilon \Delta_{nrtPS}} \eta_v}\right) \tag{12}$$

c) BE Scheduling: The BE intra-class scheduling algorithm distributes time-slots equally among BE connections. Let C be the total number of time-slots available for BE class, and $|\Delta_{BE}|$ be the number of admitted BE connections. Then the number of available slots per connection is equal to $\frac{C}{|\Delta_{BE}|}$.

Let $\varrho_w[f]$ be the bandwidth request of connection w and C_w time-slots are needed to fulfill the request. Then the BE intra-class scheduling algorithm allocates $min(C_w, C/\Delta_{BE})$ time-slots to w.

An SS with good channel conditions is able to send more data in the same number of slots, than an SS with poorer channel conditions. Therefore, equal slot distribution does not mean equal bandwidth distribution.

III. Mathematical Model

This section presents the mathematical model of TLSA. The development of an accurate mathematical model of TLSA is challenging due to the following reasons.

- Hierarchical nature of TLSA
- An rtPS packet is dropped from scheduling queues if it misses the deadline
- The bandwidth allocation by intra-class scheduling algorithms depends upon the bandwidth allocations done by the inter-class scheduling algorithm.

Therefore to make the model tractable, following assumptions are made

1) Packets arrival rate is independent of the number of packets waiting in the queue or served by the system
2) All queues are of infinite size
3) Each queue is processed in FIFO order
4) Packets arrive randomly and independent of each other

Assumption 4 implies that the packet arrival follows Poisson distribution, and the service time follows negative exponential distribution. Therefore, a single queue can be modeled as an M/M/1 system, while all scheduling queues as whole forms an M/M/C system.

Symbol	Name of Symbol	Parameter	State Set
N	Queue	N : the maximum allowed number of waiting tasks	{0,1,...,N}
μ	Exponential Server	μ: the mean rate of service for the occupied server	{ℜ}
λ	Poisson Arrival Element	λ: the mean rate of packet arrival	{}
	Merge	None	{}
	Blocker	None	{}

Fig. 1. Semantics of state variables and parameters

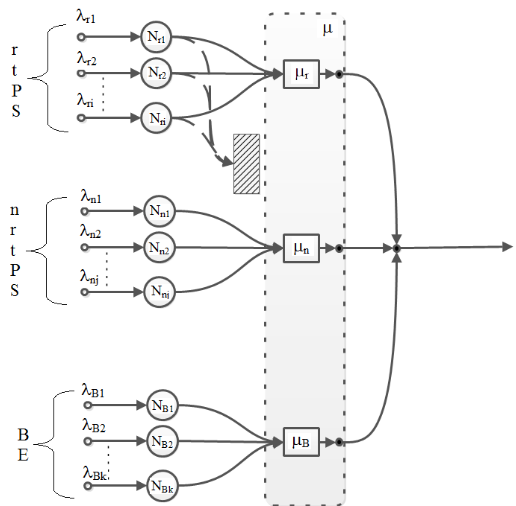

Fig. 2. Queueing model of inter-class scheduling algorithm

Therefore, the probability of 'z' arrivals during a MAC frame can be determined by Equation 13.

$$P(z) = \left(\frac{\lambda\gamma}{z!}\right) e^{-\lambda z} \tag{13}$$

The distribution of bandwidth by the inter-class scheduling algorithm can be represented by the network diagram shown in Figure 2. The figure is based on the semantics proposed by VL Wallace and R. Rosenberg [10]. The semantic of the symbols is presented in Figure 1. In the figure, μ represents the average service rate of the uplink packet scheduler. μ_r, μ_n, and μ_b denote the service rates of rtPS, nrtPS, and BE intra-class schedulers, respectively. An approximate operation of inter-class and intra-class scheduling algorithms can be logically presented by the network diagram shown in Figure 3. In the figure, μ_{ij} represents the service rate observed by connection ij.

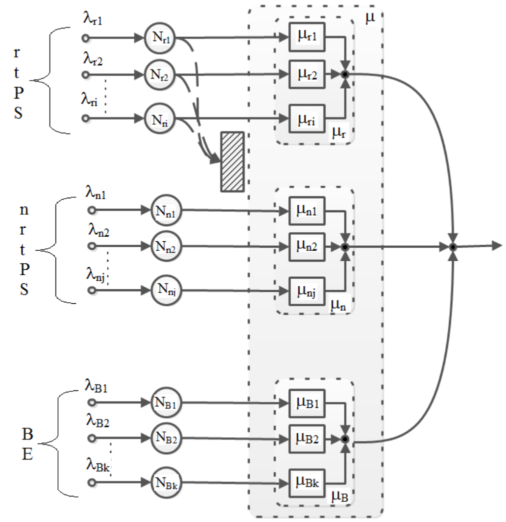

Fig. 3. Queuing model of inter-class and intra-class scheduling algorithms

A. Analysis of BE Intra-class Scheduling Algorithm

Let us assume that there are $\mid \Delta_{BE} \mid$ BE connections competing for uplink bandwidth. Connection i has an average packet arrival rate of λ_i. The MAC frame has a length of γ seconds, and ϑ_i^f is the average number of packets of connection i served per frame. This implies that the total number of frames per second is equal to $\frac{1}{\gamma}$ and thus the average number of packets served per second (ϑ_i) for connection i can be given as

$$\vartheta_i = \frac{\vartheta_i^f}{\gamma} \tag{14}$$

Let T_i be the average time to serve a packet of connection i. Then by using Equation 14, T_i can be determined as

$$T_i = \frac{1}{\vartheta_i} = \frac{\gamma}{\vartheta_i^f} \tag{15}$$

Based on equations 14 and 15, various parameters of the M/M/1 model can be determined as follow.

Scheduler utilization (ρ_i) by connection i = $\lambda_i T_i$

$$\rho_i = \frac{\lambda_i \gamma}{\vartheta_i^f} \tag{16}$$

The average response time (T_i'') by the scheduler is equal to $\frac{T_i}{1-\rho_i}$

$$T_i'' = \frac{\gamma}{\vartheta_i^f - \lambda_i \gamma} \tag{17}$$

The average waiting time (T_i') for a packet of connection i in queue is equal to $\frac{\rho_i T_i}{1-\rho_i}$

$$T_i' = \frac{\lambda_i \gamma^2}{\vartheta_i^f \left(\vartheta_i^f - \lambda_i \gamma\right)} \tag{18}$$

By using Little's formula [6], the average number of packets waiting (w_i) in data queue of connection i is equal to $\frac{\rho_i^2}{1-\rho_i}$

$$w_i = \frac{\lambda_i^2 \gamma^2}{\vartheta_i^f \left(\vartheta_i^f - \lambda_i f\right)} \tag{19}$$

The average number of packets in the system (r_i) is given by $\frac{\rho_i}{1-\rho_i}$

$$r_i = \frac{\lambda_i \gamma}{\vartheta_i^f - \lambda_i \gamma} \tag{20}$$

Standard deviation of r_i (σ_{r_i}) is equal to $\frac{\sqrt{\rho_i}}{1-\rho_i}$

$$\sigma_{r_i} = \frac{\sqrt{\lambda_i \gamma \vartheta_i^f}}{\vartheta_i^f - \lambda_i \gamma} \tag{21}$$

Another important parameter to determine is the probability ($P(\kappa)$) of κ packets waiting in the queue. The parameter is useful in determining the queue size for which the probability of overflow is below a given threshold $P(\kappa)$. That is the size of queue should be κ to avoid overflow with a probability of $P(\kappa)$.

$$P(\kappa) = 1 - \rho_i^{1+\kappa}$$

Taking log on both sides and rearranging, we get

$$\kappa = \frac{\ln(1 - P(\kappa))}{\ln(\frac{\lambda_i \gamma}{\vartheta_i^f})} - 1 \tag{22}$$

B. Analysis of nrtPS Intra-class Scheduling Algorithm

Let r_n be the bandwidth available for nrtPS class and ϱ_u be the backlog of connection u at the start of current frame. The bandwidth allocated to connection u is given by Equation 12. If ϱ_u is used as basis of bandwidth distribution, then the resulting model is very complex and demands an iterative solution. Since ϱ_u is directly proportional to λ_u, therefore to make the analysis simple, equation 12 can be rewritten as equation 23.

$$\Theta_u = min\left(\varrho_u, \alpha_u\right) + min\left(r_n, \sum_{v \epsilon \Delta_{nrtPS}} \lambda_v\right)\left(\frac{\lambda_u}{\sum_{v \epsilon \Delta_{nrtPS}} \lambda_v}\right) \tag{23}$$

To simplify the presentation of subsequent equations we assume that $\pi_u = min\left(\varrho_u[f], \alpha_u\right)$ and $\nu_u = min\left(r_f, \sum_{v \epsilon \Delta_{nrtPS}} \lambda_v\right)\left(\frac{\lambda_u}{\sum_{v \epsilon \Delta_{nrtPS}} \lambda_v}\right)$. In the simplified form, equation 23 can written as

$$\Upsilon_u = \pi_u + \nu_u \tag{24}$$

Let l_u be the average packet size of connection u, then we have

$$T_u = \frac{l_u}{\pi_u + \nu_u} \tag{25}$$

$$\rho_u = \frac{\lambda_u l_u}{\pi_u + \nu_u} \tag{26}$$

$$w_u = \frac{(\lambda_u l_u)^2}{(\pi_u + \nu_u)(\pi_u + \nu_u - \lambda_u l_u)} \tag{27}$$

$$T_i' = \frac{\lambda_u l_u^2}{(\pi_u + \nu_u)(\pi_u + \nu_u - \lambda_u l_u)} \tag{28}$$

$$r_u = \frac{\lambda_u l_u}{\pi_u + \nu_u - \lambda_u l_u} \tag{29}$$

$$T_i'' = \frac{l_u}{\pi_u + \nu_u - \lambda_u l_u} \tag{30}$$

$$\sigma_{r_u} = \frac{\sqrt{(\pi_u + \nu_u)\lambda_u l_u}}{\pi_u + \nu_u - \lambda_u l_u} \tag{31}$$

and

$$\kappa = \left(\frac{\ln(1 - P(\kappa))}{\ln(\lambda_u l_u) - \ln(\pi_u + \nu_u)}\right) - 1 \tag{32}$$

C. Analysis of rtPS Intra-class Scheduling Algorithm

For analyzing the rtPS intra-class scheduling algorithm, a new parameter δ is introduced to account for the deadlines of rtPS packets. We define δ_i be the maximum tolerable latency of connection i. That is, if a packet k of i arrives at time a_k, then it must be scheduled between a_k and $a_k + \delta_i$ to meet the latency constraint. If the packet is not transmitted before $a_k + \delta_i$, then it is considered to be expired and therefore it is dropped from the queue. The constraint of deadline makes the analysis considerably difficult than the analyses done earlier in this section.

Let $\mid \Delta_{rtPS} \mid$ be the number of rtPS connections and Θ_{rtPS} be the average bandwidth available for the rtPS class, i.e. $\Theta_{rtPS} = \beta - \sum_{i \epsilon \Delta_{UGS} \cup \Delta_{ertPS}} \alpha_i - \sum_{j \epsilon \Delta_{nrtPS}} \alpha_j - \beta_{BE}$. The algorithm assures fairness of resource allocation by using the Equations 9 and 10. This allocation scheme implies that the average service rate of connection i can be given by Equation 33.

By using Equation 11, μ_i can be defined as follows

$$\mu_i = \frac{\lambda_i(\beta - \beta')}{\sum\limits_{j\epsilon\Delta_{rtPS}} \alpha'_j} \quad (33)$$

To make the analysis tractable, we assume that each queue can be analyzed separately with an average arrival rate of λ_i and average service rate of μ_i. Using equation 33, the values of T_i and ρ_i can be determined as follows.

$$T_i = \frac{1}{\mu_i}$$

$$T_i = \frac{\sum\limits_{j\epsilon\Delta_{rtPS}} \alpha'_j}{\lambda_i(\beta - \beta')} \quad (34)$$

Similarly,

$$\rho_i = \lambda_i T_i$$

$$\rho_i = \frac{\sum\limits_{j\epsilon\Delta_{rtPS}} \alpha'_j}{\beta - \beta'} \quad (35)$$

The arrival rate of each connection obeys the Poisson distribution, the average service time follows exponential distribution, and the deadline is deterministic. Therefore, each rtPS queue can be analyzed as an M/M/1+D system. This class of queues were analyzed by D.Y Barrer [4]. The most important parameter for the rtPS class is maximum tolerable latency. Specifically, we are interested in knowing the number of packets dropped due to expiry of deadline. According to D.Y. Barrer, the packet loss probability (Q) under statistical equilibrium can be computed by Equation 36.

$$Q_i = \frac{(1-\rho_i)e^{\mu_i\delta_i(\rho_i-1)}}{1-\rho_i e^{\mu_i\delta_i(\rho_i-1)}} \quad (36)$$

Since a packet is dropped if its waiting time exceeds δ_i, therefore T''_i is always between 0 and δ_i, Mathematically, $0 < T''_i \leq \delta_i$. Since λ_i is the arrival rate and Q_i is the loss probability, therefore the average packet loss rate (ϖ_i) is the product of λ_i and Q_i and the throughput is equal to $\lambda_i l_i(1-Q_i)$

$$\varpi_i = \lambda_i Q_i \quad (37)$$

and the total packet drop rate for the entire rtPS class can be given as

$$\varpi = \sum_{j\epsilon\Delta_{rtPS}} (\lambda_j Q_j) \quad (38)$$

The average throughput (τ_i) of connection i is given by equation 39

Parameter	Value
Total uplink bandwidth	1 Mbps
Frame duration	20 ms
TDD downlink duration	10 ms
MAC propagation delay	1 μs
Cyclic prefix	8.0
Input queue size	50000 bytes
Antenna model	omni antenna
Sampling factor	8/7
Shadowing model	constant
Shadowing mean	4.0 dB
Temperature	290K
Noise factor	10.0
Service flow timeout	15 s
Transmit power	20 dBm
Receive power threshold	205e-12
Carrier sense power threshold	0.9 * Receive power threshold
Handover RSS trigger	-78.0 dBm
Propagation limit	-111.0 dBm

TABLE I. IMPORTANT SIMULATION PARAMETERS FOR COMPARATIVE ANALYSIS

$$\tau_i = \lambda_i l_i(1 - Q_i) \quad (39)$$

and the average throughput for the entire rtPS class can be determined by Equation 40.

$$\Phi = \sum_{j\epsilon\Delta_{rtPS}} \lambda_j l_j(1 - Q_j) \quad (40)$$

The probability that the average queue size is equal to κ packets at statistical equilibrium can be determined by Equation 41.

$$P_i(\kappa) = \lambda_i P_i(0) \prod_{j=1}^{\kappa} (\mu_i + C_i(j))^{-1} \quad (41)$$

Where,

$$P_i(0) = \frac{1}{\mu_i T_i + 1}$$

and

$$C_i(j) = \frac{\mu_i(\mu_i\delta_i)^{j-1}e^{\mu_i\delta_i}}{\int_0^{\mu_i\delta_i} t^{j-1}e^{-t}dt}$$

IV. Results and Discussion

Simulations were performed to assess the validity of the analytical model presented in Section III. The simulations of TLSA were performed in Qualnet 5.0 [9]. Qualnet is a commercial simulator developed by Scalable Network Technologies (SNT). It is a suite of tools to model large scale networks. The advantage of using Qualnet is that it provides a validated and faithful simulation model of the real world IEEE 802.16 networks.

The values of important simulation parameters are shown in Table I. Each experiment was repeated 50 times and the mean value is used for the comparative study. A unique pseudo-random seed was used for each repetition to alter the characteristics of simulation such as traffic generation patterns, interference levels, back-off timers and mobility patterns.

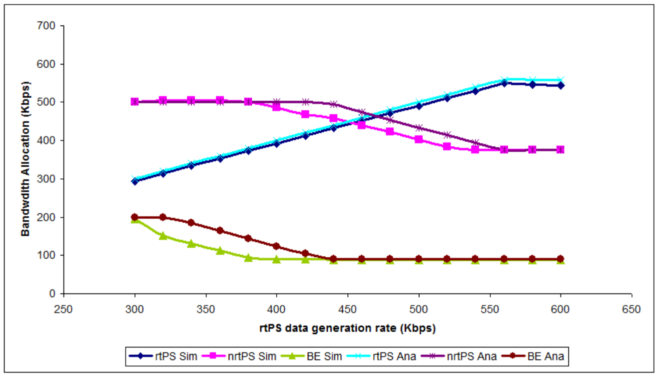

Fig. 4. Comparative analysis of inter-class bandwidth allocation

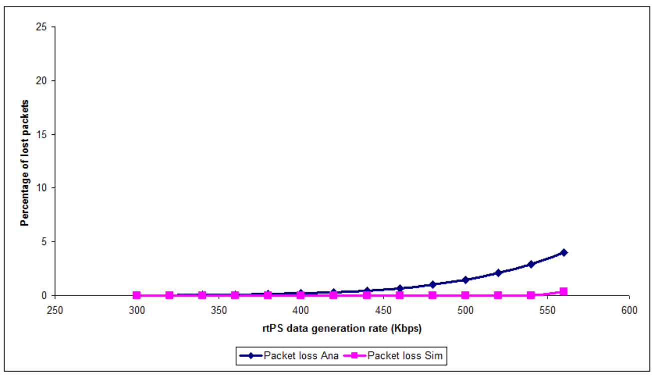

Fig. 5. Comparative analysis of packet lost in inter-class scheduling

A. Bandwidth Distribution by Inter-Class Scheduling Algorithm

The experiment was performed to compare the analytical and simulation models of inter-class scheduling algorithm. In this experiment, BE traffic was generated at an average rate of 200 Kbps. The value of β_{BE} was set to 90 Kbps. The MRTR and average traffic rate of nrtPS class were 375 Kbps and 500 Kbps, respectively. The experiment was performed with increasing load of rtPS traffic. Initially rtPS traffic was generated at an average rate of 300 Kbps, which was gradually increased to 600 Kbps.

The minimum traffic rate of rtPS traffic was set to 300 Kbps, while the maximum tolerable latency was set to 160ms. Figure 4 presents the comparison of bandwidth distribution by the analytical and simulation models. For rtPS class, the average difference between simulation and analytical results is 8.34 Kbps with a standard deviation of 2.49 Kbps.

The simulation and analytical curves of nrtPS class follow the same trend. The average difference is 12.65 Kbps with a standard deviation of 16.27 Kbps. The analytical and simulation curves of BE class show greater differences at lower rtPS traffic rates. However, the curves converge and become identical at rtPS traffic rate of 440 Kbps and greater. The average difference between the curves is 16.69 Kbps with a standard deviation of 21.25 Kbps.

The comparative analysis of percentage of lost packets are shown in Figure 5. Initially both curves follow exactly the same trend. However, deviation of the analytical curve from the simulation curve can be seen at rtPS traffic rate greater than 460 Kbps. The average difference between the curves is 0.93 percent point (pp) with a standard deviation of 1.14pp.

The comparative study suggests that the analytical model predicts the behavior of inter-class scheduling algorithm with good accuracy. Therefore, it could be concluded that the model faithfully captures the working of inter-class scheduling algorithm.

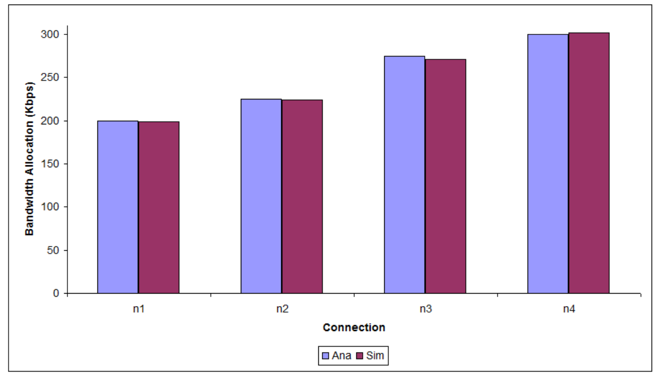

Fig. 6. Comparative analysis of nrtPS intra-class scheduling

Connection	MRTR (Kbps)	Average Traffic Rate (Kbps)
$n1$	140	200
$n2$	200	225
$n3$	225	275
$n4$	250	300
Total	815	1000

TABLE II. INPUT TRAFFIC PARAMETERS FOR COMPARATIVE ANALYSIS OF NRTPS INTRA-CLASS SCHEDULING ALGORITHM

B. Intra-Class Scheduling Algorithms

1) nrtPS Class: The comparison of analytical and simulation models of nrtPS intra-class scheduling algorithm is shown in Figure 6. For the comparison four nrtPS connections, with parameters as shown in Table II, were used. The figure reveals that the simulation results closely matches the analytical results. The average difference is 1.09 Kbps with an standard deviation of 2.42 Kbps.

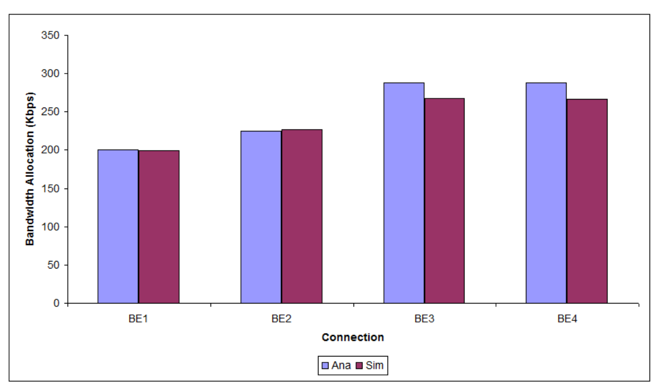

Fig. 7. Comparative analysis of BE intra-class scheduling

Connection	MRTR (Kbps)	MSTR (Kbps)	Tolerable Delay (ms)
A	400	900	40
B	100	300	60
C	200	400	60
D	300	500	80

TABLE III. INPUT TRAFFIC PARAMETERS FOR COMPARATIVE ANALYSIS OF RTPS INTRA-CLASS SCHEDULING ALGORITHM

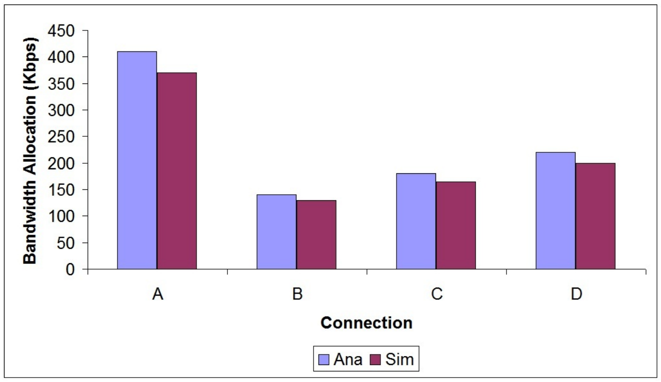

Fig. 8. Comparative analysis of rtPS intra-class scheduling

2) BE Class: The comparison of analytical and simulation models of BE intra-class bandwidth allocation is shown in Figure 7. In this experiment, four SS with one BE connection each were used. The average traffic rate of connections BE1, BE2, BE3, and BE4 were 200 Kbps, 225 Kbps, 275 Kbps, and 300 Kbps, respectively. The average difference between simulation and analytical results is 9.83 Kbps with a standard deviation of 12.14 Kbps, and therefore the models are in good agreement with each other.

3) rtPS Class: The comparison of simulation and analytical results of rtPS intra-class bandwidth allocation and associated service ratios are presented in Figures 8 and 9, respectively. In this experiment, four rtPS connections, with transmission characteristics as shown in Table III, were used. The average bandwidth allocation difference between two models is 35 Kbps with a standard deviation of 17 Kbps.

The comparative analysis provided in this section reveals that the working of inter-class and intra-class scheduling algorithms is modeled with good accuracy by the proposed analytical model. The simulation results are in accordance with the analytical model and thus validates its accuracy.

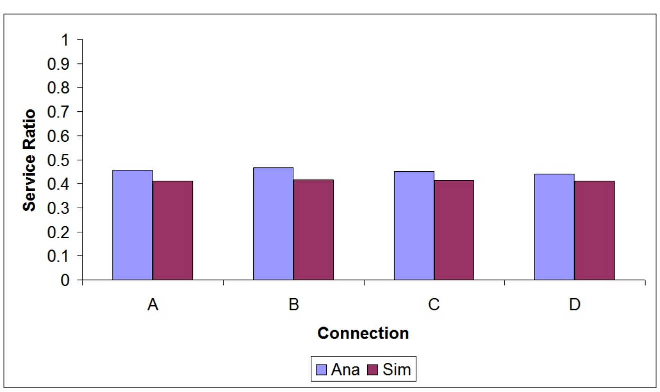

Fig. 9. Comparison of connection service ratios

V. CONCLUSION

In this paper, we presented a Markov chain based analytical model of Two Level Scheduling Algorithm (TLSA) for WiMAX networks. In TLSA an inter-class scheduling algorithm distributes bandwidth among various WiMAX service classes, while intra-class bandwidth distribution for each class is done by a class-specific scheduling algorithm. The analyses provide detailed models of both inter-class and various intra-class scheduling algorithms. The performance study allows one to understand and analyze TLSA and the simulation results reveal that the model correctly depicts the operations of TLSA. Therefore, it can be used to predict the behavior of TLSA with good accuracy as shown in section IV.

REFERENCES

[1] Zeeshan Ahmed. *Quality of Service in WiMAX for Multimedia Services*. PhD thesis, Universite de Nantes France, 2013.

[2] Zeeshan Ahmed and Salima Hamma. Efficient and fair scheduling of rtPS traffic in IEEE 802.16 point-to-multipoint networks. In *4th Joint IFIP/IEEE wireless and mobile networking conferece*, Oct. 2011.

[3] Zeeshan Ahmed and Salima Hamma. Two-level scheduling algorithm for different classes of traffic in WiMAX networks. In *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2012 International Symposium on*, pages 1–7. IEEE, 2012.

[4] DY Barrer. Queuing with impatient customers and indifferent clerks. *Operations Research*, 5(5):644–649, 1957.

[5] S.A. Filin, S.N. Moiseev, M.S. Kondakov, A.V. Garmonov, Do Hyon Yim, Jaeho Lee, Sunny Chang, and Yun Sang Park. QoS-Guaranteed Cross-Layer Transmission Algorithms with Adaptive Frequency Subchannels Allocation in the IEEE 802.16 OFDMA System. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 11, pages 5103 –5110, june 2006.

[6] BV (Boris Vladimirovich) Gnedenko and IN Kovalenko. *Introduction to queueing theory*. Birkhauser, 1989.

[7] IEEE 802.16e. IEEE 802.16-2005, IEEE standard for local and metropolitan area networks - Part 16: Air interface for fixed and mobile broadband wireless access systems amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands, February 2006.

[8] Lidong Lin, Weijia Jia, and Wenyan Lu. Performance Analysis of IEEE 802.16 Multicast and Broadcast Polling based Bandwidth Request. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 1854 –1859, march 2007.

[9] Qualnet simulator (version 5.02) http://web.scalable-networks.com/content/qualnet.

[10] Victor L Wallace and Richard S Rosenberg. Markovian models and numerical analysis of computer system behavior. In *Proceedings of the April 26-28, 1966, Spring joint computer conference*, pages 141–148. ACM, 1966.